

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**



The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

Phame

Dated 10 April 2003

THIS PAGE BLANK (USPTO)

The
Patent
Office

21FEB03 E786861-1 D00611 1/77
P01/7700 0.00-0303977.3

Patents Act 1977
Rule 16

THE PATENT OFFICE
L
21 FEB 2003
NEWPORT

Request for grant of a patent

The Patent Office
Concept House
Cardiff Road
Newport
South Wales NP10 8QQ

1.	Your reference	GB920020080GB1		
2.	Patent application number (The Patent Office will fill in this part)	0303977.3	21 FEB 2003	
3.	Full name, address and postcode of the or of each applicant (underline all surnames)	INTERNATIONAL BUSINESS MACHINES CORPORATION Armonk New York 10504 United States of America		
	Patents ADP number (if you know it)			
	If the applicant is a corporate body, give the country/state of its incorporation	State of New York United States of America	00519634001	
4.	Title of the invention	A Method and System for Generating a Tutorial Application		
5.	Name of your agent (if you have one)	C J LING		
	"Address for Service" in the United Kingdom to which all correspondence should be sent (including the postcode)	IBM United Kingdom Limited Intellectual Property Department Hursley Park Winchester Hampshire SO21 2JN		
	Patents ADP number (if you know it)	919 007		
6.	If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and (if you know it) the or each application number	Country	Priority App No (if you know it)	Date of filing (day/month/year)
7.	If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date or the earlier application	No of earlier application	Date of filing (day/month/year)	

8. Is a statement of inventorship and of right to grant of a patent required in support of this request? (Answer 'Yes' if:
a) any applicant named in part 3 is not an inventor, or
b) there is an inventor who is not named as an applicant, or
c) any named applicant is a corporate body.)
- Yes

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form

Description 7
Claim(s) 2 *DMC*
Abstract 1
Drawing(s) 5 *5*

10. If you are also filing any of the following, state how many against each item.

Priority documents -
Translations of priority documents -
Statement of inventorship and right to grant of a patent (Patents Form 7/77) 1 ✓
Request for preliminary examination and search (Patents Form 9/77) -
Request for substantive examination (Patents Form 10/77)
Any other documents (please specify) -

11. I/We request the grant of a patent on the basis of this application

C. J. Ling

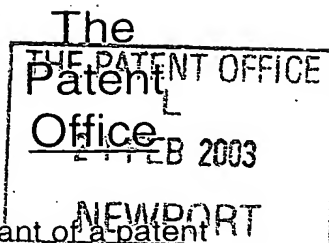
Signature

20 February 2003
Date

12. Name and daytime telephone number of person to contact in the United Kingdom C J LING
01962 815212

Patents Act 1977

Rule 15



Statement of inventorship and of right to grant of a patent

The Patent Office
Concept House
Cardiff Road
Newport
South Wales NP10 8QQ

-
1. Your reference GB920020080GB1
-
2. Patent application number
(if you know it) 0303977.3 21 FEB 2003
-
3. Full name of the or of each applicant INTERNATIONAL BUSINESS MACHINES CORPORATION
-
4. Title of invention A Method and System for Generating a Tutorial Application
-
5. State how the applicant(s) derived the right
from the inventor(s) to be granted a patent By virtue of employment and agreement
-
6. How many, if any, additional Patents Forms
7/77 are attached to this form?
-

-
7. I/We believe that the person(s) named over the page (and
on any extra copies of this form) is/are the inventor(s) of the
invention which the above patent application relates to.

C. J. Ling
Signature

20 February 2003
Date

-
8. Name and daytime telephone number of
person to contact in the United Kingdom C J LING
Tel: 01962 815212

Enter the full names, addresses and postcodes of the inventors in the boxes and underline the surnames

Gary Douglas CHAPMAN
(UK resident)
c/o IBM United Kingdom Limited
Intellectual Property Law
Hursley Park
Winchester
Hampshire SO21 2JN
England

0 857140800

Patents ADP number (if known)

Patents ADP number (if known)

If there are more than three inventors, please write their names and addresses on the back of another Patents Form 7/77 and attach it to this form

REMINDER

Have you signed the form?

Patents ADP number (if known)

A METHOD AND SYSTEM FOR GENERATING A TUTORIAL APPLICATIONField of the Invention

5 The invention relates to the field of source code development and in particular a method and system for generating a tutorial application from one or more source code elements.

Background of the Invention

10 Technology is constantly changing at a rapid pace and it is difficult for many companies to keep up to date with the latest technological developments. In order for companies to compete with their competitors in the market place it is vital that companies keep ahead of the latest
15 technological developments and provide education to their employees to enable the employees to develop the required skills.

Educating employees is a time consuming and expensive task. It takes a considerable amount of time on behalf of a course tutor in preparing course
20 materials and collating the information required. In the IT industry this is particularly a problem when employees need to be able to learn a new programming language, or how to apply a programming language to achieve a particular programming task. Every time a person learns a new programming language for example the programming language C which is a registered trade
25 mark of AT&T Bell Laboratories, or Java which is a registered trade mark of Sun Microsystems Inc, each new programming language brings with it its own way of doing things; taking the example of the programming language C, it is important to learn how the pre-processor directives operate i.e. commands that give instructions to the C Pre-processor whose job it is to
30 modify the source code of a C program before it is compiled, or what libraries are available, the types of reserved words that can be used and types of standard and user-defined identifiers, variable declarations and data types.

35 A common approach to devising tutorials for learning programming skills has been to produce a tutorial document which references the relevant elements of the source code by line number in a text editor. The person learning the programming skills is then able to read the tutorial document in conjunction with the source code. If the student does not understand for
40 example, how a particular sort routine works, by referring to the line number of the source code the person can cross reference the line number to the corresponding numbered section in the tutorial document, which explains how the sort routine works.

Another approach to devising programming tutorials can be seen in web based tutorials. The web based tutorial displays a section of source code, that provides a sort routine for example. To find out how a particular element of source code works a hyperlink is provided and by clicking on the
5 hyperlink with a mouse pointer or other input device the required information is displayed. In some examples it is possible that the hyperlink launches an application that runs the compiled code in order to observe the result for example, how a sort routine sorts elements in an array.

10 All of the above examples are time consuming on behalf of the person creating the tutorial. If a web based application is being developed, the person developing the tutorial will have to have knowledge of html to create the hyperlinks and further programming knowledge to enable the
15 tutorial to operate interactively with a student. If a more paper based approach is used, then considerable time and effort is spent writing the documentation and cross referencing the written documentation with the line number of the source code.

20 Another example is CACTUS, from the University of Madrid Spain, which offers an integrated environment to design tutorial courses for interactive applications. CACTUS is specifically for generating course documentation to teach how to use a graphical user interface. The graphical user interface is described in detail using a purpose built specification language and
25 CACTUS takes the detailed description and from this generates user documentation. A drawback with this approach is the time and effort it takes in creating an interactive application.

30 Therefore an improved method and system is required for automatically generating a tutorial application based on source code written to achieve a particular task.

Disclosure of the Invention

35 In accordance with the present invention there is now provided a method for generating a tutorial application linked to one or more source code elements, the method comprising the steps of: receiving user input indicating one or more source code elements to be selected and one or more data elements to be tagged to one or more selected elements of the source
40 code elements; tagging one or more selected source code elements with one or more of the data elements; generating the tutorial application linked to one or more source code files from said tagged source code elements; and displaying the generated tutorial application, the tagged source code elements and the data elements. Not all the source code in a source code

file may be relevant to the tutorial application that needs to be created. The user is therefore able to select just the elements of source code needed to teach a particular task.

5 Preferably the present invention provides for a method wherein the selected source code elements are tagged by a markup language. This provides an advantage for a user in that it is possible to very quickly build a tutorial application starting with one or more source code files and for the same, or a different, user to reload the tutorial source code in order
10 to update/modify the tutorial in future. A further advantage is provided for the student using the tutorial application as the student is given the benefit of viewing the selected tagged source code along with the explanation text and is able to navigate through the tutorial application in a step by step approach.

15 Preferably the present invention provides for a user creating one or more tutorial steps. This provides an advantage in that a user is able to describe and structure the tutorial application in a logical and hierarchical manner, as a tutorial can be made up of many different parts.

20 Preferably the present invention provides for a method wherein support for one or more programming languages, and one or more execution environments, is provided. The present invention is not tied into one particular programming language but provides support for many different types.

25 Preferably the present invention provides for a method wherein the data elements comprise an explanation text for the selected source code.

30 According to another aspect of the present invention there is provided for a system for generating a tutorial application linked to one or more source code files, the system comprising: means for receiving user input indicating one or more source code elements to be selected and one or more data elements to be tagged to one or more selected elements of the source code elements; means for tagging one or more selected source code elements
35 with one or more data elements; means for generating the tutorial application linked to one or more source code files from said tagged source code elements; means for displaying the tutorial application, the tagged source code elements and the data elements.

40 According to another aspect of the present invention there is provided for a computer program product directly loadable into the internal memory of a digital computer, comprising software code portions for performing the steps of any one of claim 1 to claim 6 when said product is run on a computer.

Brief Description of the Drawings

The invention will now be described by way of example only, with reference to the accompanying drawings, in which:

Figure 1, illustrates the tutorial generation application as running on a data processing system, according to a preferred embodiment of the present invention;

Figure 2, illustrates the steps that the tutorial generation application of Figure 1, carries out, according to a preferred embodiment of the present invention;

Figure 3, illustrates the input interface of the tutorial generation application of Figure 1, according to a preferred embodiment of the present invention;

Figure 4, illustrates an element of source code tagged with a markup language as it would be stored in the tutorial generation application of Figure 1, according to a preferred embodiment of the present invention; and

Figure 5, illustrates the generated tutorial application as used by a student according to a preferred embodiment of the present invention.

Detailed Description of the Preferred Embodiments of the Invention

Figure 1, illustrates the tutorial generation application 105 running on a data processing system 100. The tutorial generation application could equally run on a server which is accessed by a client device across a network. The tutorial generation application 105 comprises a development environment 130 which further comprises a source code editor 110 for developing or editing source code, a compiler 120 for translating the source code into object code, a debugger 115 for detecting errors in the source code and a display interface 125 to display the generated tutorial application. The tutorial generation application 105 can also be provided as a 'plug in' option to a preferred programming environment. A user can either create the source code in their preferred programming environment and then invoke the plugin in order to launch the tutorial generation application 105 or create the source code in the development environment provided by the tutorial generation application. The tutorial generation application 105 provides support for the many different types of programming languages and for the different types of execution environments that it will be run on. These preferences can all be manually configured from a menu option within the tutorial generation application.

By selecting the 'file menu' and then selecting 'new' it is possible to create a new tutorial. The user is prompted to enter a name for the new tutorial for example 'Thread Management'.

Referring to figure 2 at step 200, the tutorial generation application receives input from a user on the source code elements to be selected and the data elements to be tagged to the selected source code elements at step 205. Steps 200 and 205 of figure 2, will now be explained further, by way of an example with reference to figure 3.

The user can select the source code 305 in the source code editor 300 by means of an input device such as a mouse. Either by right clicking with the mouse, or by selecting an option from the tutorial generation application's 'Create' menu, or by clicking a smarticon, an input window called 'Tutorial Step' is displayed 310.

For each tutorial it is possible to have many tutorial steps. For example in the tutorial 'Thread Management' there are many steps that a programmer must write when managing threads. Each of these individual steps must be explained to a student in order for the student to understand the complexities of thread management. In such a case it is therefore necessary to select each element or group of elements of source code from one or more source code files that are required in a particular step and tag each element or group of elements with the appropriate data elements. Equally the tutorial 'Thread Management' could be part of a larger tutorial for example 'Asynchronous Beans' and as such the user is able to describe and structure the tutorial in a logical hierarchical approach when creating the tutorial for the first time or by modifying an existing tutorial.

The input window 310 provides a means in which to input the data elements which will be tagged to the source code. Firstly, the tutorial step is given a title 315; in this example the title is 'Initialise WorkManager'. The tutorial step is given a sequence number 320, in this example the number given is the number 1 which dictates that this particular tutorial step is the first step in a sequence. Subsequent steps would be labelled 2, 3, 4 and so on. Lastly, the data elements that are to be tagged to the selected source code elements are entered into the input box 325. In this example the data elements that comprise the explanation text that will be generated by the tutorial generation application. 105 explain how the WorkManager is initialised. When all the required information is input into the tutorial step window 310, the user can select the 'OK' button 330 to continue or equally the user can select the cancel button 335 to disregard the information input into the input boxes 315, 320 and 325 of the tutorial step window 310.

Referring back to figure 2, at step 215 the data elements are tagged to the selected source code. The tags are provided by a markup language such as XML. XML is a universal format for structured documents on the world wide web. XML is a set of rules for designing text formats that allow metadata to be structured using tags (words bracketed by '<' and '>') and attributes (of the form name = "value") to delimit the metadata. Step 215 of figure 2, will now be explained further, by way of an example with reference to figure 4 which illustrates a selected element of source code tagged using XML as the markup language and tagged with the attributes of the input data elements as shown in figure 3.

Firstly, the selected source code is tagged by the tutorial generation application 105 with a tutorial step tag 400 with attributes of the sequence number 405 in this example the sequence number is '1' which was input into the tutorial step sequence input box 320 in figure 3 and the title of the tutorial step 410 as input into the tutorial step title input box 315 in figure 3. The tutorial step tag 400 flags to the tutorial generation tool 105 precisely where the selected tutorial step begins and where it ends 435. A code segment tag 415 marks where the selected source code 305 begins and ends with the code segment close tag 420. Finally the explanation text tag 425, tags the data elements that comprise the explanation text as input into the explanation text input box 320 in figure 3 and flags where the data elements finish by the explanation text closing tag 430. Referring back to steps 200, 205 and 215 of figure 2, these steps are repeated until all the tutorial steps have been completed.

Referring back to figure 2 at step 220 the tutorial generation application 105 generates a tutorial application from the tagged source code in figure 4. The tutorial application is displayed at step 225 along with the selected source code. The display interface of step 225 of figure 1, will now be explained further, by way of an example with reference to Figure 5.

Figure 5 depicts the display interface of the tutorial generation application 105. The interface comprises a code window 510, an explanation window 530, a list of the tutorial steps 500 contained within the tutorial application, in this example the tutorial application 'Thread Management' and a results window 520. Although Figure 5 depicts a display interface comprising the components above, it will be appreciated by a person skilled in the art that numerous configurations of the above components are possible.

The tutorial steps that were compiled in steps 200 and 205 of figure 1 are displayed in a list 505, which enables a student to select each element within the list 500 such that the student can select which part of the

5 tutorial that they would like to explore. Each time the student selects a
tutorial step to explore from the list 505, for example by selecting
'Initialize WorkManager' 500, the selected source code that is titled
'Initialize WorkManager' is displayed in the code window 515. The tutorial
application 'step explanation' generated from the tagged source code is
10 displayed in the explanation window 530. The student is able to read the
source code in the code window 515, whilst referring to the explanation
window 530 for the explanation text. The tutorial is therefore self
contained without the student needing to refer to further documentation to
15 explain the fundamental concepts.

A results window 535 allows the student to run the tutorial application by
selecting the 'run example' option from the list 505. The tagged source
code is executed within the results window 535 and the executed source code
15 is displayed corresponding to the steps selected by the student from the
list 505.

CLAIMS

1. A method for generating a tutorial application linked to one or more source code elements, the method comprising the steps of:

receiving user input indicating one or more source code elements to be selected and one or more data elements to be tagged to one or more selected elements of the source code elements;

tagging one or more selected source code elements with one or more of the data elements;

generating the tutorial application linked to one or more source code files from said tagged source code elements; and

displaying the generated tutorial application, the tagged source code elements and the data elements.

2. A method as claimed in claim 1 wherein the selected source code elements are tagged by a markup language.

3. A method as claimed in claim 1 wherein support for one or more programming languages is provided.

4. A method as claimed in claim 1 wherein support for one or more execution environments is provided.

5. A method as claimed in claim 1, wherein receiving user input further comprises creating one or more tutorial steps.

6. A method as claimed in claim 1 wherein the data elements comprise an explanation text for the selected source code elements.

7. A system for generating a tutorial application linked to one or more source code files, the system comprising:

means for receiving user input indicating one or more source code elements to be selected and one or more data elements to be tagged to one or more selected elements of the source code elements;

means for tagging one or more selected source code elements with one or more data elements;

BEST AVAILABLE COPY

means for generating the tutorial application linked to one or more source code files from said tagged source code elements;

means for displaying the tutorial application, the tagged source code elements and the data elements.

8. A system as claimed in claim 7 wherein means for the selected source code elements are tagged by a markup language.

9. A system as claimed in claim 7 wherein means for support for one or more programming languages is provided.

10. A system as claimed in claim 7 wherein means for support for one or more execution environments is provided.

11. A system as claimed in claim 7, wherein receiving user input further comprises means for creating one or more tutorial steps.

12. A system as claimed in claim 7 wherein the data elements comprise means for an explanation text for the selected source code.

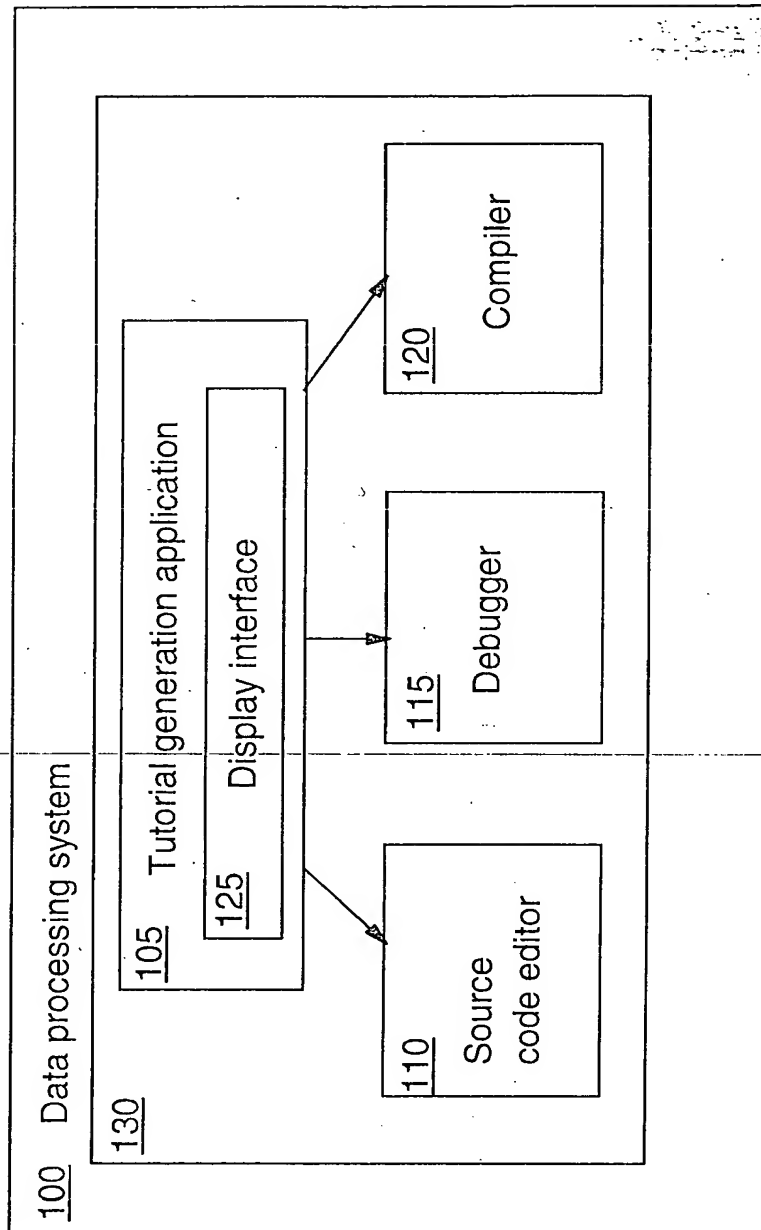
13. A computer program product directly loadable into the internal memory of a digital computer, comprising software code portions for performing the steps of any one of claim 1 to claim 6 when said product is run on a computer.

A METHOD AND SYSTEM FOR GENERATING A TUTORIAL APPLICATIONABSTRACT

5 The invention relates to the field of source code development and in particular a method and system for generating a tutorial application from one or more selected source code elements. The invention provides for a user to select one or source code elements and one or more data elements to be tagged to one or more source code elements. The selected source code
10 elements are tagged using a mark-up language and a tutorial application is generated displaying the tagged selected source code elements and the data elements.

BEST AVAILABLE COPY

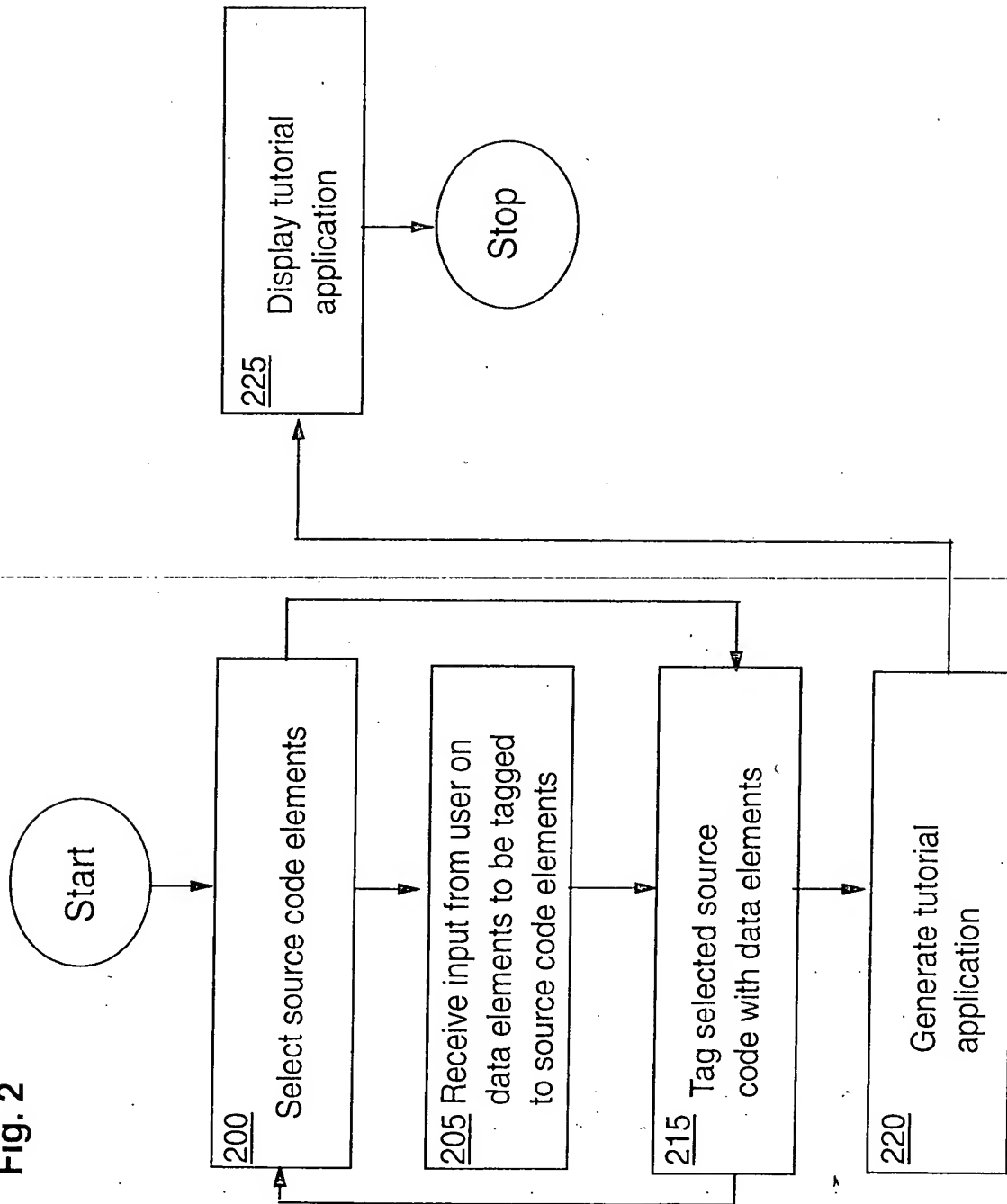
Fig 1





THIS PAGE BLANK (USPTO)

Fig. 2



THIS PAGE BLANK (USPTO)

Figure 3 is a screenshot of a Java IDE showing a Java file named 'AsynchronousWorkManager.java'. The code defines a package 'com.example', imports 'java.util.concurrent.*', and defines a 'public class AsynchronousWorkManager' with a 'public void startWork()' method. The method uses 'ThreadPoolExecutor' to create a thread pool and 'Future' objects to manage asynchronous tasks. Annotations 300-335 point to various parts of the code and the IDE interface.

Annotations and their corresponding elements:

- 300: Points to the package declaration 'package com.example;'.
- 305: Points to the import statements 'import java.util.concurrent.*;'.
- 310: Points to the class declaration 'public class AsynchronousWorkManager;'.
- 320: Points to the 'startWork()' method signature.
- 325: Points to the 'ThreadPoolExecutor' constructor call.
- 330: Points to the 'Future' object creation.
- 335: Points to the 'Future' object's 'get()' method call.

The code snippet shown is:

```

package com.example;

import java.util.concurrent.*;

public class AsynchronousWorkManager {

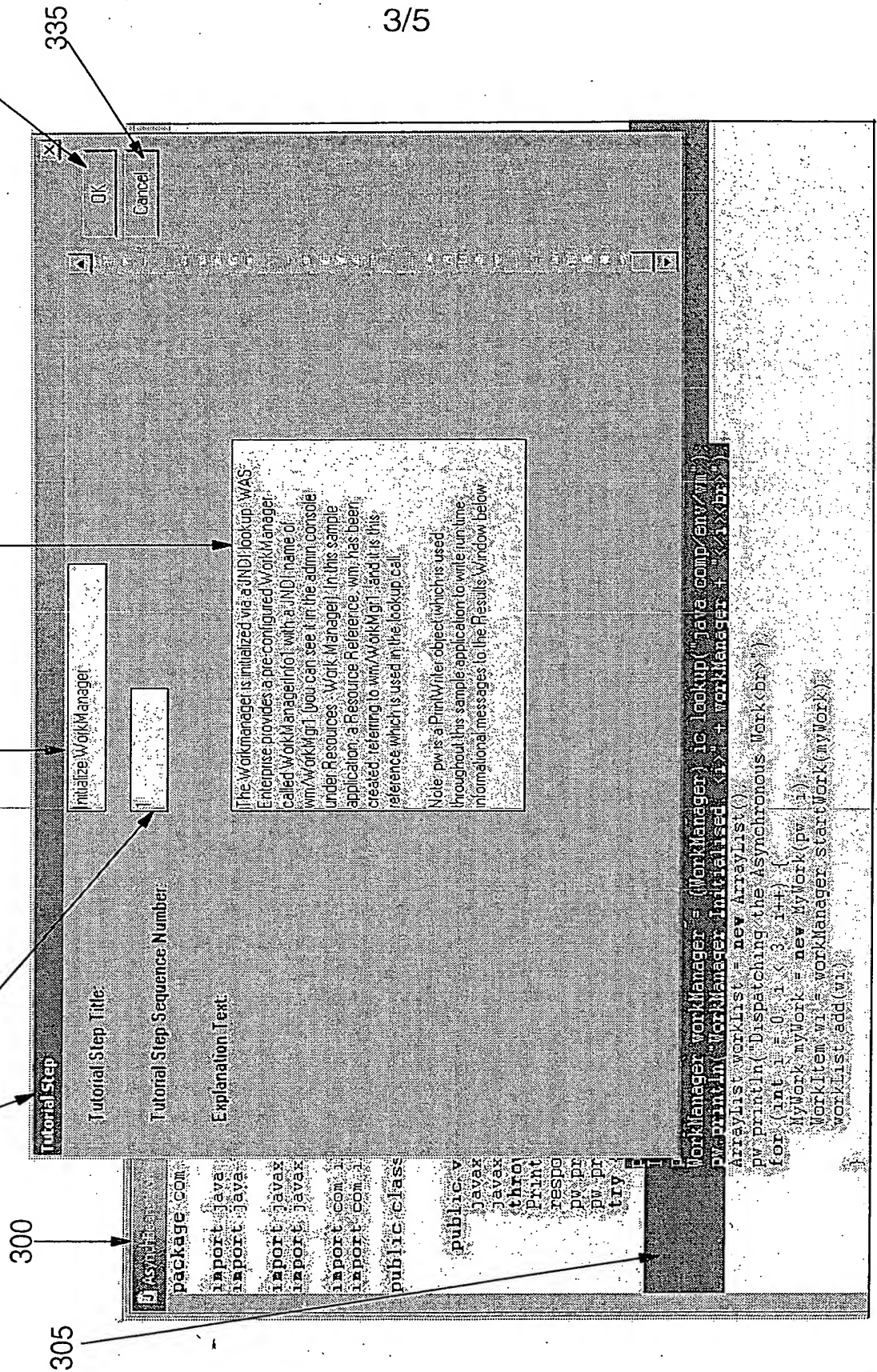
    public void startWork() {
        // Create a thread pool
        ThreadPoolExecutor executor = new ThreadPoolExecutor(
            10, 10, 1, TimeUnit.SECONDS,
            new SynchronousQueue<>());

        // Create a Future object
        Future<String> future = executor.submit(
            new Callable<String>() {
                @Override
                public String call() throws Exception {
                    // Perform some work
                    return "Hello, World!";
                }
            });

        // Get the result
        String result = future.get();

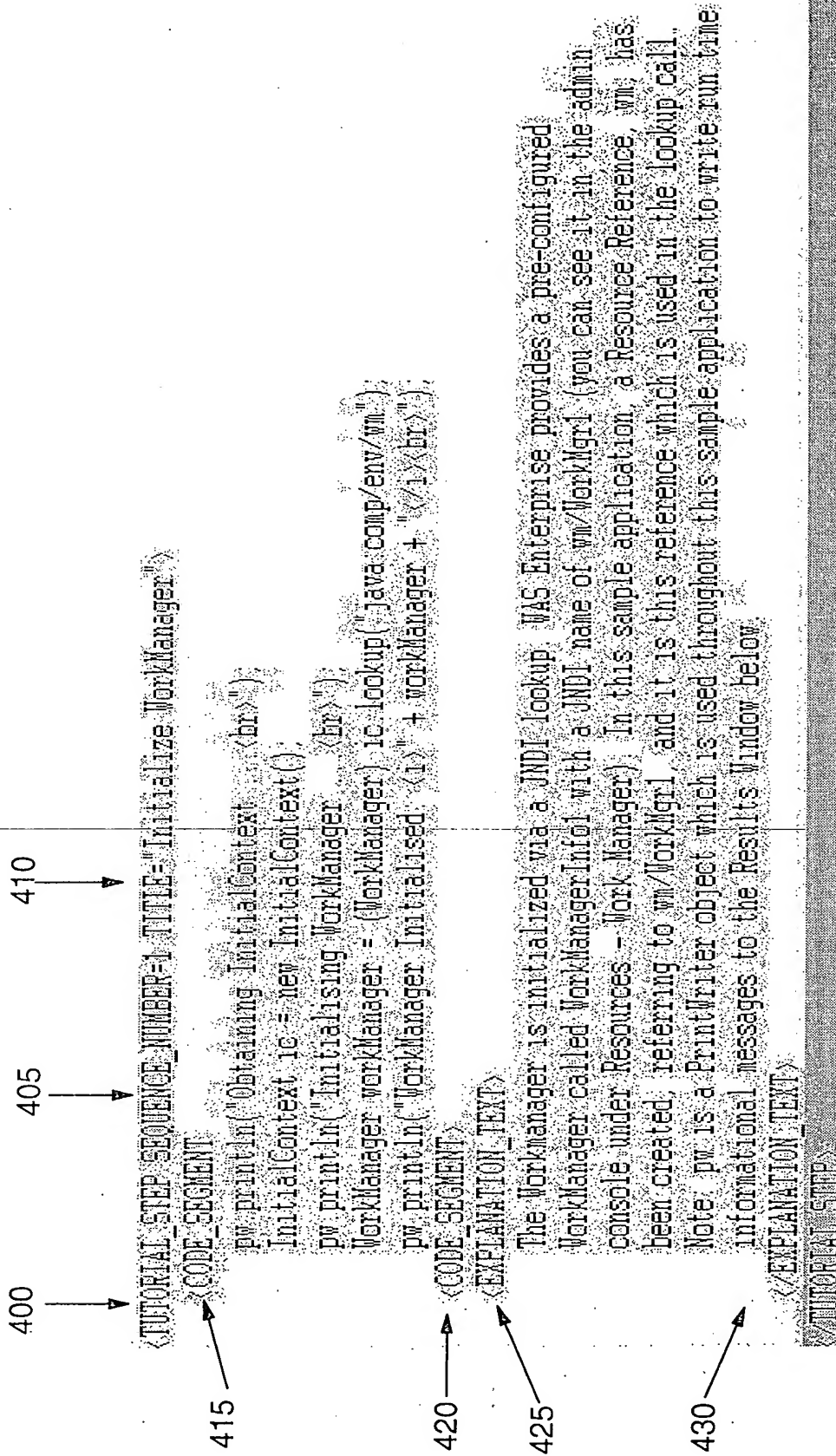
        // Print the result
        System.out.println(result);
    }
}

```



THIS PAGE BLANK (USPTO)

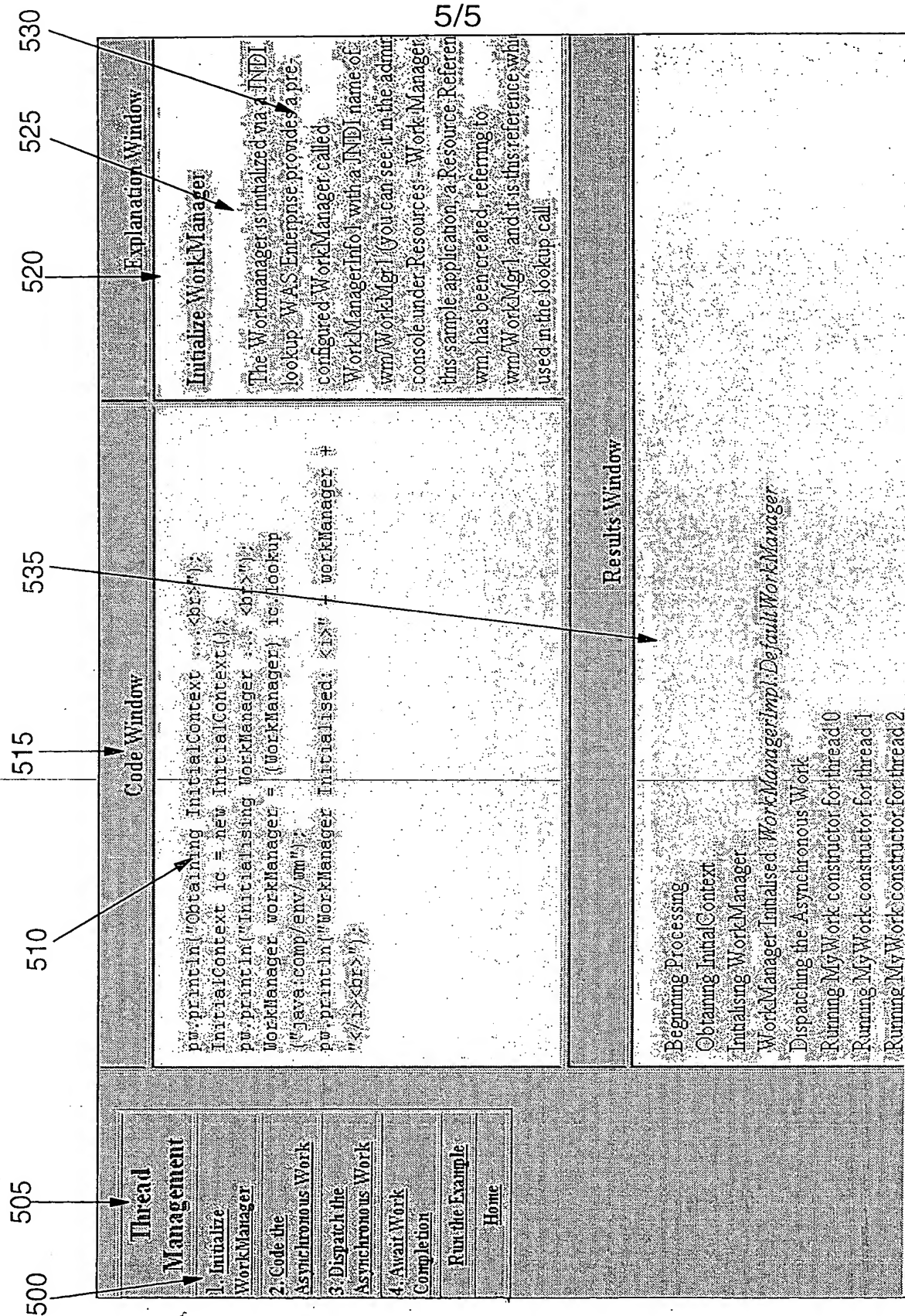
Fig. 4





THIS PAGE BLANK (USPTO)

Fig. 5



THIS PAGE BLANK (USPTO)